HEWLETT-PACKARD
PRIVATE

CONTROLLER PROCESSOR
EXTERNAL
REFERENCE SPECIFICATIONS

July 7, 1971
Stan Mintz
Cliff Wacken

Project #2046

INTRODUCTION

The design of complex peripheral device controllers, in general, requires elaborate "control" Logic to operate efficiently. Contained herein is a description of a Controller Processor, which can be used in a variety of device controllers.

The processor is contained on one printed circuit card (approximately 11" x 13").

This document is intended for use on two levels. The reader desiring general information will find it contained in the first paragraph of each section; however, if the reader requires sufficient information to design a controller, using this processor, he will find that such information is contained in subsequent paragraphs and the appendix.

To facilitate the compact design MSI and 3-state bussing is used extensively. The major sections of the processor are a 16-bit Arithmetic Logic Unit (ALU), 6 general storage registers, a decrementable counter, rotate/shift logic, and all the necessary branching and control logic necessary to implement the instruction set.

TABLE OF CONTENTS

PHYSICAL DESCRIPTION

A. GENERAL

The two inputs to the ALU are the A and B Buses. These buses are driven only by the internal general registers, the counter and the Immediate Operand, and are not available to the outside. (See Fig. 1.) The A bus is driven by the Immediate Operand, the counter, and 3 of the general purpose registers. The B bus is driven by 4 of the general purpose registers. One register is common to both buses. The output of the ALU is latched into a holding register, which serves to hold the data for either output or storage into a general register. The output of this holding register is fed into the Rotate/Shift Logic. The output of the Rotate/Shift Logic is the MI/O Bus. This bus is 3 State and is used for both input/output and the return path to the registers. The instruction set contains both arithmetic and logical instructions along with conditional branching. Also provided are sixteen flag lines that permit selective branching on external conditions.

B. INPUT/OUTPUT BUS

The output of the rotate shift logic is driven by HP104A 3-state driver gates capable of sinking or sourcing 40 ma. Of this 40 ma, 24 ma are available to the user. The user should terminate the bus commensurate with the additional loading he requires. The data on the bus is ground true. If it is desired to have the bus go to a known state when the drivers are in the 3rd state, the user should terminate such as to provide that state. (The only termination recommended is "up", forcing a logic zero bus.) Data may only be put on the bus by the user thru HP104A

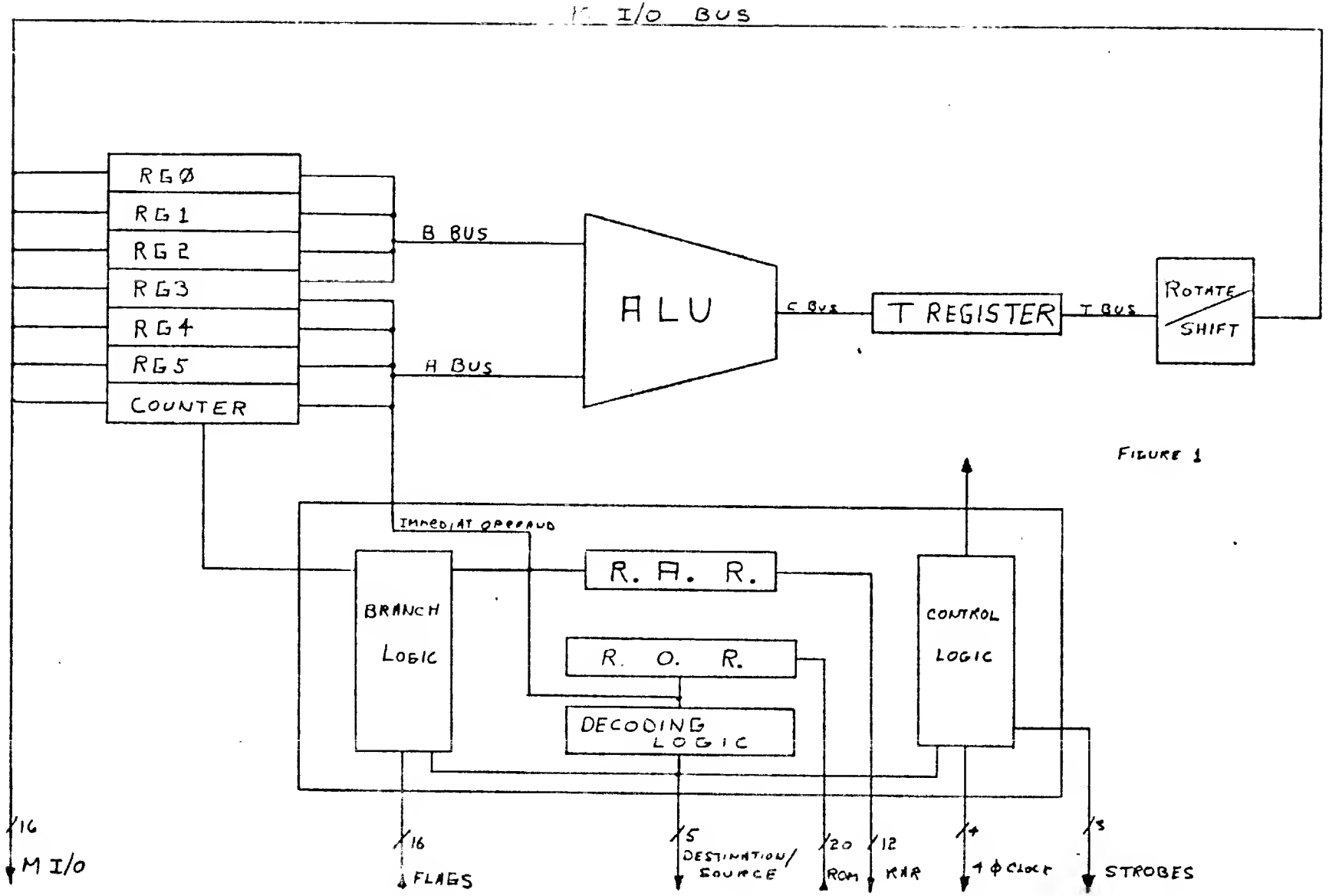

I/O BUS
RGØ
RG1
RG2
RG3
RG4
RG5
COUNTER
B BUS
A BUS
ALU
C BUS
T REGISTER
T BUS
ROTATE / SHIFT
IMMEDIAT OPERAND
R. A. R.
BRANCH LOGIC
R. O. R.
DECODING LOGIC
CONTROL LOGIC
16 M I/O
16 FLAGS
5 DESTINATION/ SOURCE
20 ROM
12 RAR
4 φ CLOCK
5 STROBES

FIGURE 1

Driver gates, and then only when an input strobe is present. This strobe appears during the last 1/4 of the cycle and is Positive true. All of the buses have the capability of only being partially selected. That is, the option of selecting the upper byte, lower byte, or the whole word is provided.

C. REGISTERS

1. General Purpose - There are six 16-bit working registers designated, RGØ through RG5. Each register can be loaded from the MI/O Bus, and can drive the A and/or B Bus. That is, it is possible to write into any one of the registers from the MI/O Bus while placing the contents of two onto the A and/or B Bus. RGØ through RG3 are connected to the B Bus, while RG3 through RG5 are connected to the A. RG3 is connected to both buses. It should be noted here that the data being written into the registers may be from any source connected to the MI/O Bus. It is also possible to write into, or read from, the upper or lower byte independently from one another.

The registers are implemented with latches. Data being presented to them should be present at least 50 ns before the end of the cycle and must remain for at least 10 ns past the end of the cycle. After an I/O reset or Power Up sequence, the state of these registers will be all bits logical true (all ones).

2. T-Register - The T-register is a 16 bit latch. It serves as the master for the general-purpose registers which act as slaves during recirculate instructions. Also, it is the output register for ouput instructions.

The T-Register is loaded mid-cycle. Sixty ns after the loading the data is valid on the MI/O Bus. In general, the data will always

be valid during the last 1/4 of the cycle and it is during this time that the output strobes (an upper byte and lower byte strobe is provided) appear. The Input strobe is exactly 1/4 cycle in duration and is positive true.

3. Counter - A 16 bit binary counter capable of being preset from the MI/O Bus, counted down, and gated onto the A Bus is available. The counter may be decremented in every instruction. The counter can be used as a 7th register when it is not in use as a counter. It is possible to branch to the contents of the counter, thus, computed addressing is possible (described later). The counter can be decremented until a zero state is reached. It will then remain in the zero state until initialized again.

If a preset and a decrement are programmed in the same instruction, the preset will prevail. It is illegal to decrement if the contents of the counter is being used as the next address. (The mnemonic for decrementing is DEC and for presetting is CTM. To gate this counter onto the A Bus CTA is called.)

4. SAVE Register - A 12 Bit register is provided to save the current address +1, thus providing single level jump to subroutine capability.

5. ROM Address Register (RAR) - The RAR is a 12 bit register capable of being incremented. It can also be preset from one of 4 sources:

a) The Decrementing Counter.
b) The Save Register
c) The Branch Field in the Instruction Word
d) The Address selected on the maintenance panel when it is connected.

The RAR feeds directly into the ROM, to address the word desired.

6. ROM Output Register (ROR) - The ROR is a 20 bit register holding the current word from the ROM. At the beginning of each cycle the new ROM word is clocked into the register. The output is directed into the Branch, Control and Decoding Logic.

D. ARITHMETIC AND LOGIC UNIT

The arithmetic and Logic Unit is capable of Providing the following functions:

FUNCTION	MNEMONIC
A PLUS B	ADD
A • B	AND
A + B	IOR
A ⊕ B	XOR
A MINUS B	SUB
$\overline{A}$	CMA
$\overline{B}$	CMB
A	PSA
B	PSB
(IMM.OP.) PLUS B	ADI
(IMM.OP.) • B	ANI
(IMM.OP.) + B	IOI
(IMM.OP.) ⊕ B	XOI
(IMM.OP.)	PSI

The output of the ALU is the input to the T-register. As the rotate/shift logic follows the T-register it is possible to perform any of the above operations and then shift the result.

E. ROTATE/SHIFT LOGIC

The rotate/shift logic is such that it is possible to call a rotate or shift left one, right one, or left four. The rotates are full word operations having no meaning if used in conjunction with a byte instruction.

The mnemonic are:

Rotate left one	-	RL1
Rotate right one	-	RR1
Shift left one	-	SL1
Shift right one	-	SR1
Rotate left four	-	RL4
Shift left four	-	SL4

No rotate or shift operation allows the data to pass straight through.

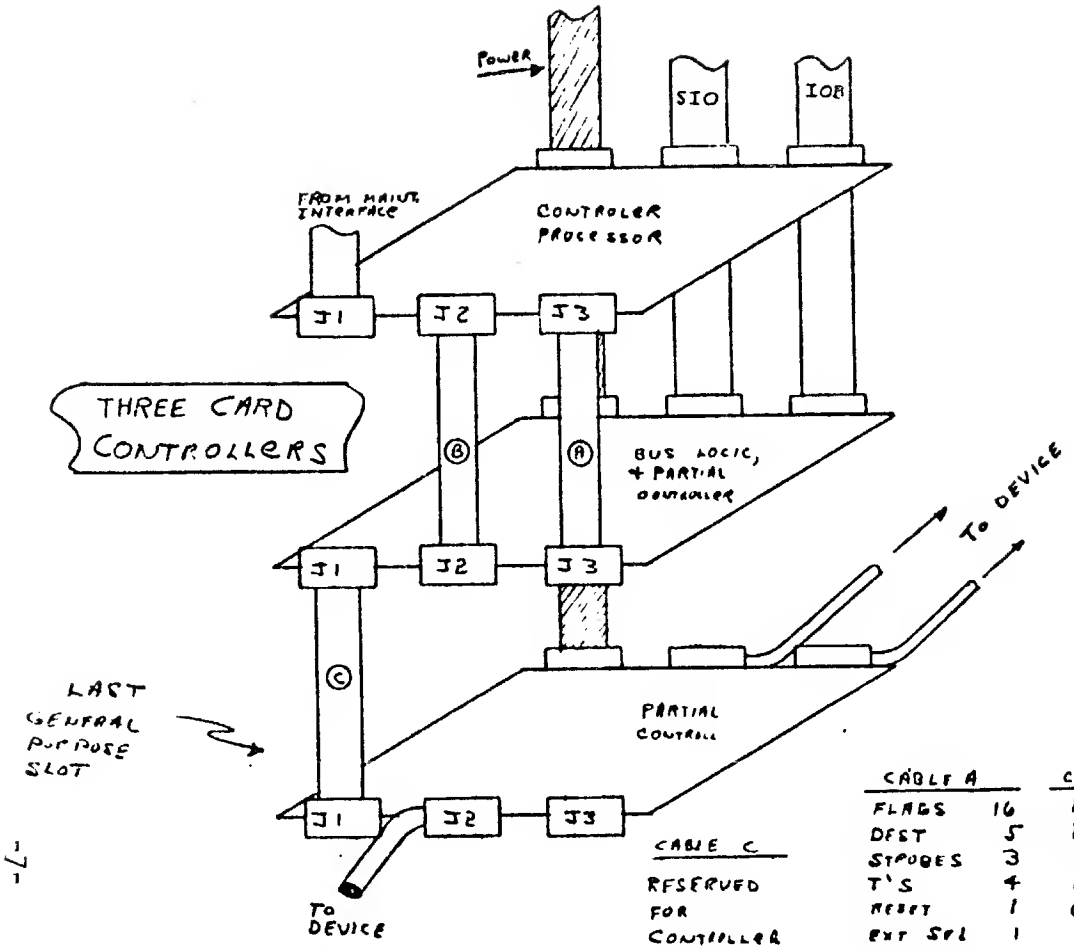
F. INTERNAL CLOCK

The processor cycle time can be selected to best fit the need of the controller. The speed is determined by what crystal is selected - crystals up to 10 MHz may be used. The cycle consists of four evenly divided phases, each phase being 100ns when the processor is run at maximum clock rate.

The user has available to him 4 clock lines. Each of the lines are one of the phases. These lines are all ground true. The processor uses the trailing edge of these lines for timing and it is recommended that the user also use the trailing edge when trying to synchronize with the processor.

G. ROM

The Read Only Memory is contained on the users portion of the controller. A suggested configuration is shown in Appendix C. The memory should be organized on a modular basis capable of being increased in increments of 256 words up to a 4K page. (It is possible to further increase the size of memory by utilizing a paging scheme.) The cable configuration necessary to allow addressing and returning of instructions is shown in Figure 2 (Cable B).



CABLE A	
FLAGS	16
DEST	5
STROBES	3
T'S	4
RESET	1
EXT SEL	1
MIO	16
GNDS	4
	50

CABLE B	
ROM	20
RAR	12
ROMFAB	1
GNDS	16
	50

CABLE C

RESERVED FOR CONTROLLER INTRA-COMMUNICATION

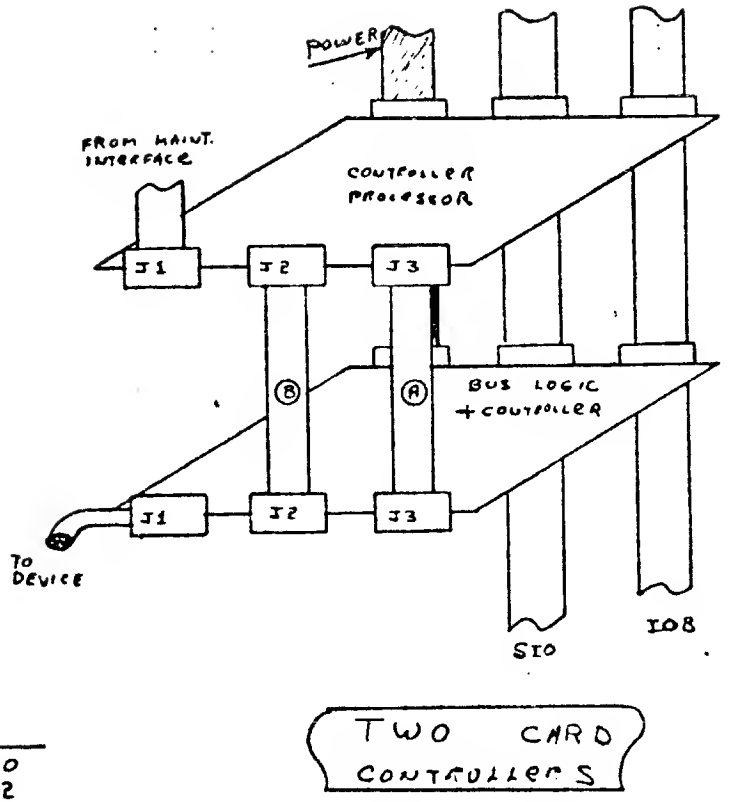


FIGURE 2

H. MISC.

Direct branching is possible by testing one of sixteen available flag lines. These lines are positive true.

Also available is an external register select line which may be used to select an External Register in place of any of the general purpose registers. This line is ground true, and is used in conjunction with T3.
Five destination source bits are provided to select the source or destination desired when exercising in Input/Output Instruction. These lines are Positive True and are Valid only when one of the strobes is true.

Reset can be accomplished from the maintenance panel. The Reset line is wire - "OR" able and available to the user. It is suggested that the system master reset be buffered and connected to the Processor Reset. It is ground true.

PHYSICAL ASSEMBLY

As the Controller Processor is completely housed on one card, the minimum system size can be as few as two cards. It is expected in fact that the majority of systems will be on two cards; however, for larger systems it is possible to use any number of additional cards. All of the cards will be mounted in the Alpha general I/O slots. Suggested cabling and card positioning is shown in figure 2.

SYSTEM REQUIREMENTS

The following is required of the users card(s) in the system:

1. The ROM input and output must be available to the Processor.

2. It must interface the IOB and the SIO. (It is recommended that one of the standard Bus logic interfaces be used.)

3. It must be able to disable his ROM. See Appendix C.

FUNCTIONAL DESCRIPTION

A. GENERAL

The ROM instruction word is 20 Bits wide, with the highest order 5 bits containing the op-code. The processor can be functionally described by examining these op-codes. They fall into three general categories, arithmetic/Logic Codes, input/output codes, and Branch Codes. Referencing to figures 3-5 will be helpful while reading the various descriptions.

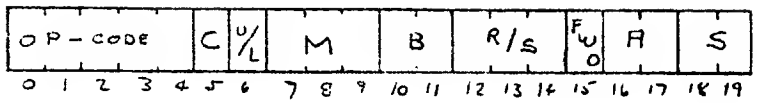
B. ARITHMETIC/LOGIC CODES

These op-codes utilize either word type 1 or 2 (see figure 3) depending on whether or not an immediate operand is required. Word Type 1 is divided into the following fields:

Counter Field (C) — This is a one bit field to determine whether the counter is to be decremented or not. A 0 in the field causes a decrement at the end of the current instruction. The mnemonic is DEC. (If no DEC is specified the assembler will place a 1 in the field.)

Upper Byte/ Lower Byte — (U/L and Full Word Override) The options here are to select the upper byte, the lower byte, or the full word. If only one byte is selected, the other byte is left totally unaffected. The mnemonics are UPH and LWH. (If no byte is specified, full word is assumed.)

WORD TYPE 1



OP-CODES
- ADD
- AND
- IOR
- XOR
- SUB
- CMA
- CMB
- PSA
- PSB

COUNTER (I)
- 0 ⇒ NOP
- 1 ⇒ COUNT

M
1 OF 8 REGISTERS OR THE COUNTER FOR STORAGE

B
1 OF 8 REGISTERS OR THE COUNTER FOR ALU INPUT SELECTION

A
1 OF 4 REGISTERS FOR SELECTING THE OTHER ALU INPUT

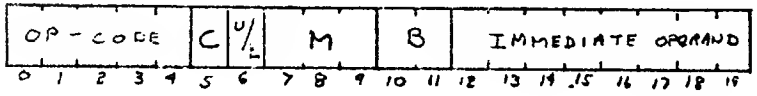
R/C
- SR1
- SL1
- RR1
- RL1
- SL4
- RL4
- PAS

U/L
SELECTS: UPPER BYTE, LOWER BYTE.

FULL WORD OVERRIDE
OVERRIDES U/L AND ALLOWS A FULL WORD ON THE BUS

S
SPECIAL UNDER USER CONTROL

WORD TYPE 2



OP-CODE
- AMI
- ANI
- IOI
- XOI
- PSI

IM. OP
ANY 8 BIT CONSTANT

FIGURE 3

M-Field

The contents of the M field determines the target register. The result of a calculation will be loaded into this target register. Any one of 8 possible ports can be selected. The "8th" port (the six registers and counter have been previously described) is the external register select. Whether or not a controller contains this register, is of course optional. The mnemonics are EXM, CTM, RØM, ... R5M. (No specification results in EXM being selected.)

B-Field

One of 4 sources to the ALU can be selected and gated onto the B Bus. The possibilities are the registers RGØ, RG1, RG2, and RG3. The mnemonics are RØB, R1B, R2B, and R3B. (If no specification is made R3B will be selected.)

A-Field

One of four sources can be selected to be gated onto the A Bus and hence into the ALU. The four sources are the counter, RG3, RG4, and RG5. The mnemonics are CTA, R3A, R4A, and R5A. (If no specification is made, R3A will be selected.)

Rotate/Shift Field

As its name implies, this field invokes the rotate or shift option desired. The mnemonics are RL1, RR1, RL4, SL1, SR1, and SL4. (If no rotate/shift is specified, the code for no action is set.)

S-Field — 2 bits are available to whatever special function the user requires. The Field is valid when the word type one line is true.

Op-code field — The op-codes of word type 1 are:

ADD (Addition) — The A bus is added to the B bus and the results placed in the T register.

SUB (Subtraction) — The B bus is subtracted from the A bus and the results placed in the T register. The subtraction is two's complement.

IOR (inclusive "OR") — The A and B buses are "OR" ed together, and the results stored in the T register.

XOR (exclusive "OR") — The A and B buses are exclusive "OR" ed together and placed in the T register.

CMA (complement A) — The complement of the A Bus is placed in the T register.

CMB (complement B) — The complement of the B Bus is placed in the T register.

PSA (Pass A) — The A Bus is placed in the T register.

PSB (Pass B) — The B Bus is placed in the T register.

Word type 2 is like word type 1 with the exception that the immediate operand replaces the Rotate/Shift field, the special field, the A field, and the Full Word override Bit. Because the immediate operand is only 8 bits wide, it is necessary that it be specified in which byte it is to be stored. The requirement for the A field is superfluous as the immediate operand is automatically placed on

the A Bus. The counter, M, and B fields are still available. The op-codes of word type 1 are:

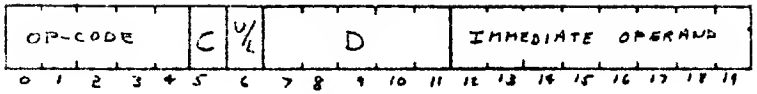
ADI (Addition to the immediate operand)	The immediate operand is added to whichever byte is selected on the B Bus and the results are placed in that byte of the T register.
ANI (AND with the immediate operand)	The immediate operand is "AND" ed with whichever byte is selected and the resulting byte placed in the T register.
XOI (Exclusive or with the immediate operand)	The immediate operand is EXCLUSIVE "OR" ed with whichever byte is selected and the resulting byte placed in the T register.
IOI - (Inclusive OR with the immediate operand)	The immediate operand is "OR" ed with whichever byte is selected and the resulting byte placed in the T register.
PSI (PASS the immediate operand)	The immediate operand is placed in the selected byte of the T register.

The general sequence of events is that the contents of the A and B Buses are fed into the ALU, then the result is stored into the T register, which in turn feeds the rotate/shift logic which finally drive the M Bus with the data to be written into the selected target register. If the counter is to be counted, it should not be used as a source that instruction.

C. INPUT/OUTPUT CODES

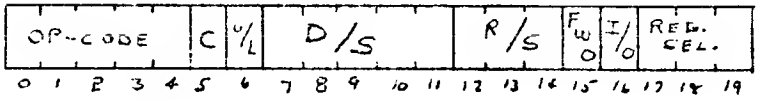
There are two input/output opcodes, OTI (output immediate operand) and IOC (input/output control), utilizing word types 3 and 4 respectively. (See Figure 4.)

Word Type 3



OP-CODE	DESTINATION
OTI	5 BITS FOR SELECTING WHICH LOCATION THE CONTENTS OF THE M/O BUS IS INTENDED FOR

Word Type 4



OP-CODE	I/O	REG. SEL.
IOC	DETERMINES IF AN INPUT OR OUTPUT INSTRUCTION	DETERMINES WHICH INTERNAL REGISTER IS BEING USED

FIGURE 4

Word type 3 contains the following fields:

Counter field	Previously described.
U/L field	Previously described.
D/S (Destination Source)	This is a five bit field to determine the intended destination for the data generated during this instruction. The Output strobes indicate when this field is valid.
Immediate Operand	Previously described.
Op-Code	OTI - The immediate operand is placed on the MI/O Bus.

Word type 4 is divided into the following fields:

Counter field	Previously described.
U/L field	Previously described.
D/S (Destination Source)	This is as described previously. Additionally, the five bits could now be interpreted as a source or a destination. Whether it is a source or a destination is determined by the I/O field. (Described below)
R/S	Previously described.
Full Word Overide	Described in Up/Low in Section B. Arithmetic/Logic Codes.
I/O (Input/Output)	This is a one bit field to determine whether an input or an output is being executed.. The mnemonics are INP and OUT.
Register Select	This is a three bit field that determines which register (or the counter) is to be selected. Whether the register is the source of the data or the target depends on the state of the I/O Bit. The mnemonics are CTI, EXI, RØI, R1I, ..., R5I, for the counter, the external register, and RGØ thru RG5 respectively.
Op-Code field IOC	The IOC either sets up path for data to be placed on the MI/O Bus from the selected register, or clocks data into the selected register from the MI/O Bus, depending on the I/O Bit.

D. BRANCH CODES

These op-codes utilize either word type 5 or 6 (see figure 5) depending on whether the branch is unconditional, occurring on internal conditions or occurring on external conditions. Word type 5 is selected for the absolute and internal branches. The fields are defined as follows:

C — Previously defined.

RCS (Reverse condition sense) — This one bit field is to complement the sense of an internal condition when a conditional branch is being executed. The mnemonic is RCS. (If no specification is made, the unaltered condition will prevail.)

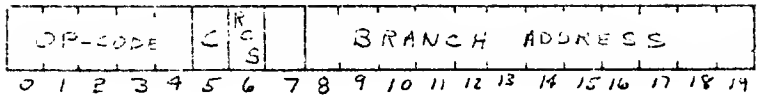
Branch Address — This is a twelve bit field to allow direct branching anywhere within the current 4K page.

OP-CODES — The op-codes of word type 5 are:

JMP (JUMP) — This is an unconditional branch to the address specified in the branch address field.
i.e. RAR ← BRANCH ADDRESS;

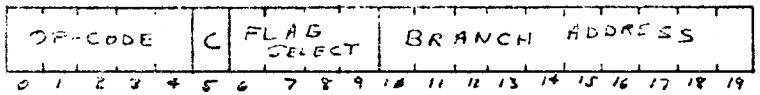
JMX (JUMP,I) — This is an unconditional branch to the address specified in the counter. The branch address field and the higher order 4 bits of the counter are ignored. (The counter must be made positive true before using as an address).
i.e. RAR(0|12) ← COUNTER(4|12);

Word type 5



OP-CODE	BRANCH ADRS
JMP (JUMP)	BRANCH ANYWHERE WITHIN 4 K.
JMX (JUMP COUNTER, I)	
JMZ (JUMP IF T=0)	
JXZ (JUMP IF COUNTER=0)	
JSZ (JUMP IF SIGN=0)	
CAL (JUMP TO SUBROUTINE)	
CAX (JUMP SUB CNT, I)	
CMZ (JUMP SUB T=0)	
CXZ (JUMP SUB CNT=0)	
JOV (JUMP IF OV)	
RMN (RETURN FROM SUB T=0)	
RXN (RETURN FROM SUB CNT=0)	
RTN	

Word type 6



OP-CODE	FLAG SEL.	BRANCH ADRS
JFS	1 OF 16 CAN BE SELECTED	CAN BRANCH ANYWHERE WITHIN CURRENT 1 K.
RFS		

FIGURE 5

JMZ (Jump on T = 0)

This is a branch to the address specified in the branch address field if the condition T = 0, is met, if not, the next instruction in sequence is executed. In all internal conditional branch instructions, the condition under test is the one that existed mid-cycle of the previous instruction, that is, the value being loaded into the T register.

```
i.e. If T = 0 then
  RAR ← BRANCH ADDRESS
else
  RAR ← RAR+1;
```

JXZ (Jump on counter = 0)

The branch is taken if the condition, counter = 0, is met, otherwise, the next instruction in sequence is executed. Care should be used as the test occurs before the counter is decremented. (Do not do a preset of the counter in one instruction and a conditional branch on Counter = 0 in the very next, assuming the test will fail as it will not - the counter was still 0 at the time of test.)

```
i.e. IF COUNTER = 0 THEN
  RAR ← BRANCH ADDRESS
else
  RAR ← RAR+1;
```

JSZ (Jump on Sign = 0)

The branch is taken if the condition, Sign = 0, is met, otherwise, the next instruction is executed.

```
i.e., IF SIGN = 0 THEN
  RAR ← BRANCH ADDRESS
else
  RAR ← RAR+1;
```

JOV (Jump on Overflow)

The branch is taken if the condition, OVERFLOW is met, otherwise the next instruction is executed.

```
i.e.,  IF OVERFLOW THEN
          RAR ← BRANCH ADDRESS
       else
          RAR ← RAR+1;
```

CAL (Call Subroutine)

This is a branch to a subroutine while retaining the return address (in the SAVE register). The subroutine must be within the current 4K Block.

```
i.e.,  BEGIN
          RAR ← RAR+1;
          SAVE ← RAR;
          RAR ← BRANCH ADDRESS
       end
```

CAX (Call Subroutine, 1)

This is a branch to a subroutine as above, except that the address of the subroutine is the lower order 12 bits of the counter.)Again, the contents must be positive true).

```
i.e.,  BEGIN
          RAR ← RAR+1;
          SAVE ← RAR;
          RAR (0|12) ← COUNTER (4|12);
       end
```

CMZ (Call Subroutine on T=0)

This is a branch to a subroutine if the condition, T = 0 is met. If it is not, the next instruction in sequence is executed.

```
i.e.   IF T = 0 THEN
          BEGIN
             RAR ← RAR+1;
             SAVE ← RAR;
             RAR ← BRANCH ADDRESS;
          end else
             RAR ← RAR+1;
```

CXZ (Call Subroutine on Counter = 0)

If the counter is zero, the call to subroutine is made, otherwise the next instruction in sequence is executed.

```
i.e.,  IF COUNTER = 0 THEN
          BEGIN
             RAR ← RAR+1;
             SAVE ← RAR;
             RAR ← BRANCH ADDRESS;
          end else
             RAR ← RAR+1;
```

RTN (Return from Subroutine)

This instruction is an absolute return from a subroutine. The next instruction executed is the one whose address is stored in the SAVE Register.

```
i.e.,  RAR ← SAVE;
```

RMN (Return on T ≠ 0)

This instruction is a conditional return from a subroutine. If the condition, T≠0, is met, the return will occur; however, if the condition fails, a branch to the instruction indicated in the branch address field will occur. The next instruction in sequence will not implicitly be executed.

```
i.e.,  IF T ≠ 0 THEN
          RAR ← SAVE
       else
          RAR ← BRANCH ADDRESS;
```

RXN (Return on Counter ≠ 0) — This instruction returns if the condition, COUNTER ≠ 0, is met, and branches if the condition is not.

```
i.e.,  IF COUNTER ≠ 0 THEN
          RAR ← SAVE
       else
          RAR ← BRANCH ADDRESS;
```

Reiterating a few points, the conditional branches or returns are always on the results of the previous instruction. When the branch is indirect, only the lower order 12 bits of the counter are used. The counter is always incremented after the test. The branch is only within the current 4K. And the instruction sequentially following a return is never implicitly executed.

If the conditional branch is on external conditions, then word type 6 is selected.

C — As previously described.

Flag Select — This 4 bit field is to determine which of the 16 possible flags is under test. Only one of the 16 flags may be under test at one time.

Branch Address

This is a 10 bit field to allow direct branching anywhere within the current 1K segment of the 4K page.

OP-CODES

JFS-(Jump on Flag Set)

This is a branch to the address specified in the Branch Address field if the specified flag is present. If the flag is not present, the next instruction in sequence is executed.

```
i.e.,  IF SELECTED FLAG = 1 THEN
          RAR ← BRANCH ADDRESS
       else
          RAR ← RAR+1;
```

RFS - (Return On Flag Set)

This instruction is similar to the conditional returns on internal conditions. If the specified flag is present, the return from subroutine is executed, if the flag is not present, the branch is taken.

```
i.e.,  IF SELECTED FLAG = 1
          RAR ← SAVE
       else
          RAR ← BRANCH ADDRESS;
```

APPENDIX A

CONTROLLER PROCESSOR ASSEMBLER

Willard Reed

July 7, 1971

APPENDIX A

CONTROLLER PROCESSOR

ASSEMBLER

INTRODUCTION

This is an assembler for the HP Controller Processor 20 bit ROM. Refer to the Controller Processor External Reference Specifications for information on the operation and structure of this unit.

GENERAL

The fields are free form except that all labels must begin in column one and only labels may begin in column one. Only columns one through 40 are used for coding; columns 41 through 80 are used for comments. The mnemonics may be entered in any order except for certain restrictions (see section on Mnemonics). If more room is needed for coding a "/" will continue to the next card.

In general if a mnemonic is not specified for a field the assembler will place all ones in that field, (exceptions are the shift/rotate field default case is 1ØØ, and full word override is cleared by an upper or lower half mnemonic). However, the first mnemonic (OP Code) must always be specified.

The labels, mnemonics and other fields are separated by one or more blanks or by a comma which may be followed by one or more blanks.

The last card in the deck must contain the END mnemonic.

LABELS

Labels are one to six alphanumeric characters the first of which must be alphabetic. Labels must start in column one. If there is no label column one must be blank.

The assembler assigns to the label a value which is equal to the address of the line containing the label; or in the case of the EQU equal to the value in the operand field.

Labels must not be duplicated and mnemonics must not be used as labels.

MNEMONICS

The first mnemonics in a line must denote an OP Code, an EQU, an SKP, or an END. The following mnemonics, octal values and label references may be in any order; except of course these items may only be used if they are appropriate for the specified OP Code.

The following is a list of legal mnemonics.

OP Codes	A Ports	B Ports	M Ports
ADD,JMP,	CTA (COUNTER)	RØB (RGØ)	CTM (COUNTER)
AND,JMX,	R3A (RG3)	R1B (RG1)	EXM (EXT. REG.)
IOR,JMZ,	R4A (RG4)	R2B (RG2)	RØM (RGØ)
XOR,JXZ,	R5A (RG5)	R3B (RG3)	R1M (RG1)
SUB,JSZ,			R2M (RG2)
CMA,CAL,			R3M (RG3)
CMB,CAX,			R4M (RG4)
PSA,CMZ,			R5M (RG5)
PSB,CXZ,			
ADI,JOV,			
ANI,RMN,			
XOI,RTN,			
PSI,JFS,			
OTI,RFS,			
IOC,NOP,			
RXN,IOI,			

Destination Bits	Flag Select	Input/Output	I/O Reg. Sel.	Reverse Jump Select
DØØ	FØØ	INP	CT1 (COUNTER)	RCS
DØ1	FØ1	OUT	RØ1 (RGØ)	
.	.		R11 (RG1)	
.	.		R21 (RG2)	
.	.		R31 (RG3)	
DØ7	.		R41 (RG4)	
D1Ø	.		R51 (RG5)	
.	F17		EX1 (EXT. REG)	
.				
.				
D37				

Counter	Upper/Lower Half	Rotate/Shift	Alternate Page	Other Mnemonics
DEC	UPH	RL1	PAG	SKP
	LWH	RR1		EQU
		RL4		END
		SL1		
		SR1		
		SL4		

Immediate Operand and Branch Address — These fields may contain either a decimal number, an octal number (preceded by "#"), a label reference, or the form *±X, where X is an unsigned octal or decimal number. When a number is used it may be unsigned or preceded with a "+" or "-" sign.

COMMENTS

Columns 41 through 80 may be used for comments. If column one contains an "*" then the whole line is considered comments:

ERRORS

The assembler will detect most errors which result in an inconsistent or inconclusive symbolic coding line. Errors may cause the assembler to lose sync between pass one and pass two. Hence, if the assembly has errors, label references may not reference correctly.

The following is a list of assembler error messages:

C1 Illegal character in a list of assembler error messages
C2 Number with value greater than 177777 octal
I2 Two mnemonics assign values to the same field
L1 Label more than six characters
L2 Label begins with illegal character
M1 Multiply defined label or mnemonic used as label
N1 A mnemonic is used which is illegal for this OP Code
S1 Label reference longer than six characters
S2 First mnemonic not OP Code, EQU, SKP or END
U1 Reference is made to an Undefined label.

GENERAL OPERATING INSTRUCTIONS

This is a two pass assembler it is used as follows:

a) Load the assembler and begin execution
b) The assembler will PAUSE (Type PAUSE and halt). Load the deck to be assembled and press run.
c) The assembler will again PAUSE. Reload the deck and press run.
d) The assembler will now print the listing and PAUSE again. Press run and the output tape will be punched. If a tape is not desired turn on switch 15 and press run. You will now be back at step b.

TO USE CONTROLLER PROCESSOR ASSEMBLER
ON CUPERTINO BOSS/DOS SYSTEM

1. To Load DOS
   A. Set switch REG to 77760B
   B. Set Loader Enabled
   C. Load Address-Preset-Run
   D. Computer will halt 10B
   E. Protect Loader
   F. Set switch REG to 00001
   G. Push RUN

2. CRT will type "INPUT FR, CO"
   Type CO CR-LF
   Type :DATE, X CR-LF

3. Make sure Line Printer is ready
   Type :JO CR-LF

4. Put deck to be assembled in card reader and ready the card reader

5. Type :ST,S,A,5 CR-LF

6. Cards will read and computer will type DONE?

7. Type YES CR-LF

8. Type :JF,A

9. If a tape is desired put SW15 down; if no tape is desired put SW15 up

10. Type :RUN, UCAM, 99 CR-LF and the program will assemble

If an I/O Device is not ready CRT will type "I/O ERR NR EQT #N"

Where N is the Device Logical Number

To recover make device ready and type :UP,6 CR-LF
:GO CR-LF

APPENDIX B

THE CONTROLLER MAINTENANCE PANEL

Cliff Wacken

July 7, 1971

The Controller Maintenance Panel consists of the Alpha Software and Hardware Maintenance Panels with overlays to change the switch-display nomenclature, the Controller Maintenance Interface Card, and interconnecting cables.

The interface card should be inserted in an I/O slot adjacent to the processor card of the desired controller. It communicates with the controller, primarily the processor, through the front connectors (J1,J2,J3) via 50 conductor flat cables. This requires additional cabling to the normal controller configuration.

The interface card connects to the two switch-display panels with two 50 conductor flat cables. Power for the two switch-display panels comes directly from the CPU power supply, and the interface card receives it's power from the I/O backplane.

The Controller Maintenance Panel will be used with controllers in an off line mode. It has the capability to halt and restart the controller ROM program at any specified address or execute one instruction at a time. It can display and change most of the processor's registers. It will also allow the use of an external clock source. Following are explanations of the displays and controls.

DISPLAYS

ROM DATA	Displays the 20 bit output of the Read Only Memory, which is the input to the processor's ROM Output Register (ROR).
ROM ADDRESS REGISTER	Displays the 12 bit output of the processor's ROM Address Register.
MIO BUS	Displays the state of the 16 bit bus that provides data communication between the controller and processor.
FLAGS	Displays the status of 16 flags at the input of the processor flag circuitry.
SELECTED REGISTER	Displays the 16 bit contents of the register selected, i.e., Counter, Register 0 - Register 5. There is no display if a register is not selected, or the processor is in the RUN mode.
REG SELECT/ NO SELECT	Indicates which one of 7 registers (including the counter) is being displayed on SELECTED REGISTER and is enabled for loading. NO SELECT indicates there is no register selected for display (SELECTED REGISTER blank) OR loading.
RUN/HALT	RUN lamp indicates the processor is cycling, HALT lamp indicates it is not.

CONTROLS

RUN/HALT

A momentary switch that changes the processor from a RUN to a HALT mode, and visa-versa.

The processor starts with the execution of the instruction displayed in the ROM register at the address displayed in the RAR register.

When the processor halts, the RAR displays the address of the next instruction to be executed, the ROM displays the next instruction to be executed, and the processor ROM Output Register (ROR) has this instruction loaded in it. The processor halts and starts after the end of T3 and before the beginnning of TØ.

SINGLE CYCLE

A momentary switch that allows one cycle of the 4 timing pulses. The cycle starts with the leading edge of TØ, proceeds through T1 and T2 and ends with the trailing edge of T3.

The instruction displayed in the ROM register, which is the contents of the address displayed in the RAR register, will be executed when the SINGLE CYCLE switch is activated.

During the cycle these two registers are updated, so at the end they display the address and instruction to be executed next.

RAR SWITCH REG/LD RAR	The RAR SWITCH REG is 12 toggle switches which work with the momentary switch, LD RAR, to change the contents of the processors' ROM Address Register The desired address is set into the RAR SWITCH REG and the LD RAR switch is then activated. The processors' ROM Address Register is changed to the pattern in the RAR SWITCH REG. The ROM display and the processors' ROM Output Register change to the contents of the new address.
RAR COMPARE REGISTER	A 12 toggle switch register which determines the ROM address for a loop or halt breakpoint, and a pulse output from the RAR COMPARE connector.
RAR COMPARE ENABLE/INHIBIT	A toggle switch which enables or inhibits the loop or halt breakpoint option.
RAR COMPARE LOOP/HALT	A toggle switch which determines whether the processor ROM program will jump to a pre-determined ROM address or halt. The option will occur when enabled by the ENABLE/INHIBIT switch, and at the ROM address set into the RAR COMPARE REGISTER.
LOOP Option	When the address in the processors' ROM Address Register (arrived at through natural progression of the ROM program) agrees with the address set in the RAR COMPARE REGISTER the processors' ROM Address Register will be forced to an address determined by the RAR SWITCH REG.

LOOP Option (Continued)

The instruction contained in the address set in the RAR COMPARE REGISTER will be completed, and the next instruction executed will be that of the address set in the RAR SWITCH REG.

HALT Option

When the address in the processors' ROM Address Register (arrived at through natural progression of the ROM program) agrees with the address set in the RAR COMPARE REGISTER, the ROM program is halted.

The instruction contained in the address set in the RAR COMPARE REGISTER will be completed. The RAR display and the ROM display will indicate the address and instruction to be executed next.

SWITCH REG

A 16 toggle switch register used to set up a data pattern that can be entered onto the MIO bus and into the processors' counter and registers (Register Ø - Register 5).

LD REG

A momentary switch that loads the contents of the SWITCH REG into the processors' counter or registers (Register Ø - Register 5) via the MIO bus.

The counter or register that is loaded must have been previously selected, and is indicated by the REG SELECT display.

The SWITCH REG is allowed on the MIO bus only when the LD REG switch is first activated.

REGISTER SELECT ⇐ ⇒

Two momentary switches used to control the selection of processor registers and the NO SELECT mode. Registers must be selected to enable loading and displaying them.

INITIALIZE

A momentary switch that controls certain processor register outputs and issues a clear pulse to the controller.

The processor's ROM Output Register, ROM Address Register, and Save Register are forced to logical "Ø". The processors' Register Ø - Register 3 and T-Register are forced to a logical "1". The processor counter is not changed.

EXT CLOCK/ INT CLOCK

A toggle switch which will inhibit the processor clock in the EXT CLOCK position, and allow use of an external clock. The switch should be in the INT CLOCK position for normal operation.

EXT CLOCK

A BNC connector which is a clock input to the processor from an external source.

The input is terminated with 50Ω to ground. The external source should provide a square wave excursion from ground to +2.5V to +5V (TTL compatible). Four external clock periods will equal one processor cycle time.

RAR COMPARE OUTPUT

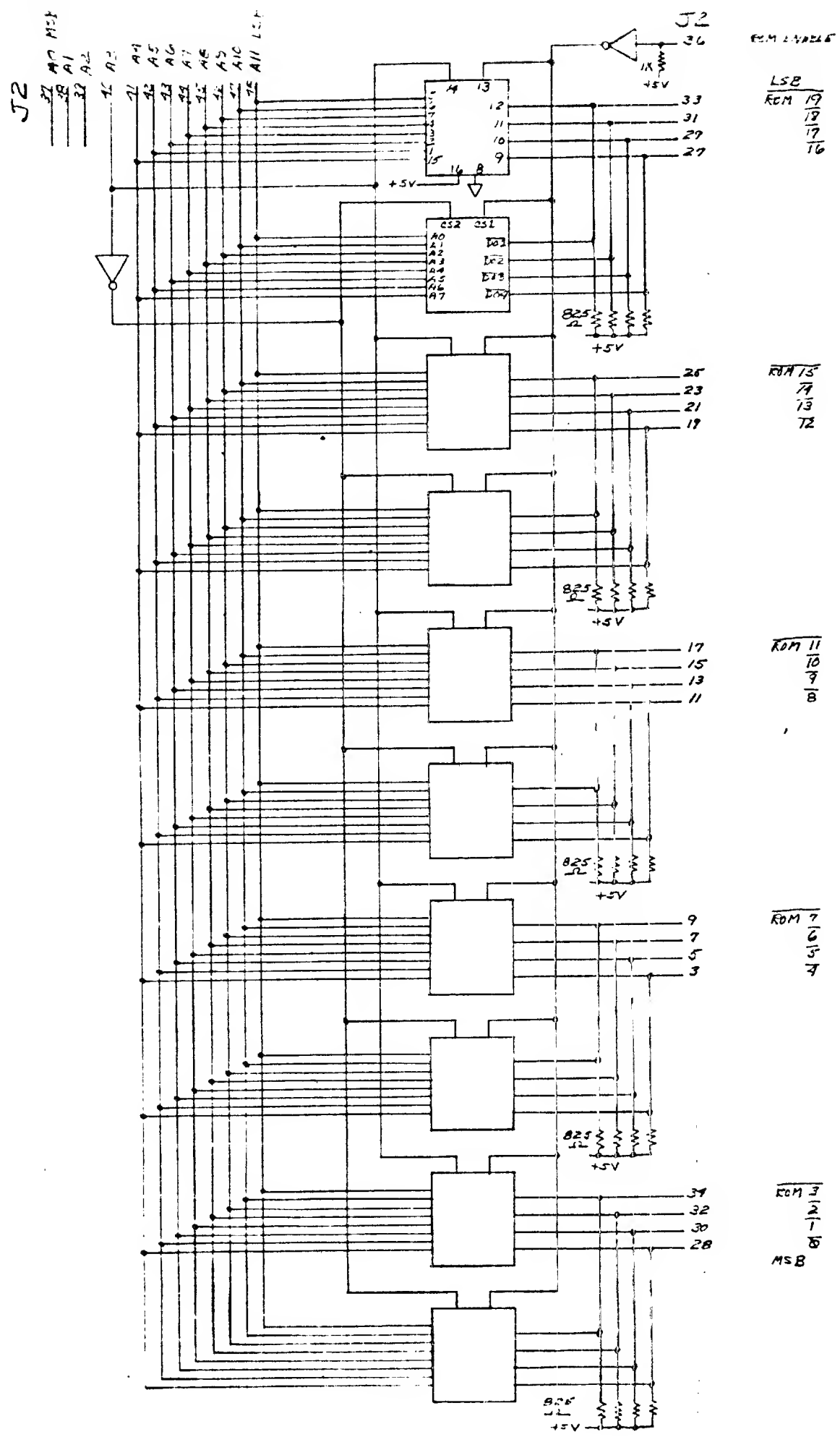
A BNC connector providing a pulse for triggering purposes.

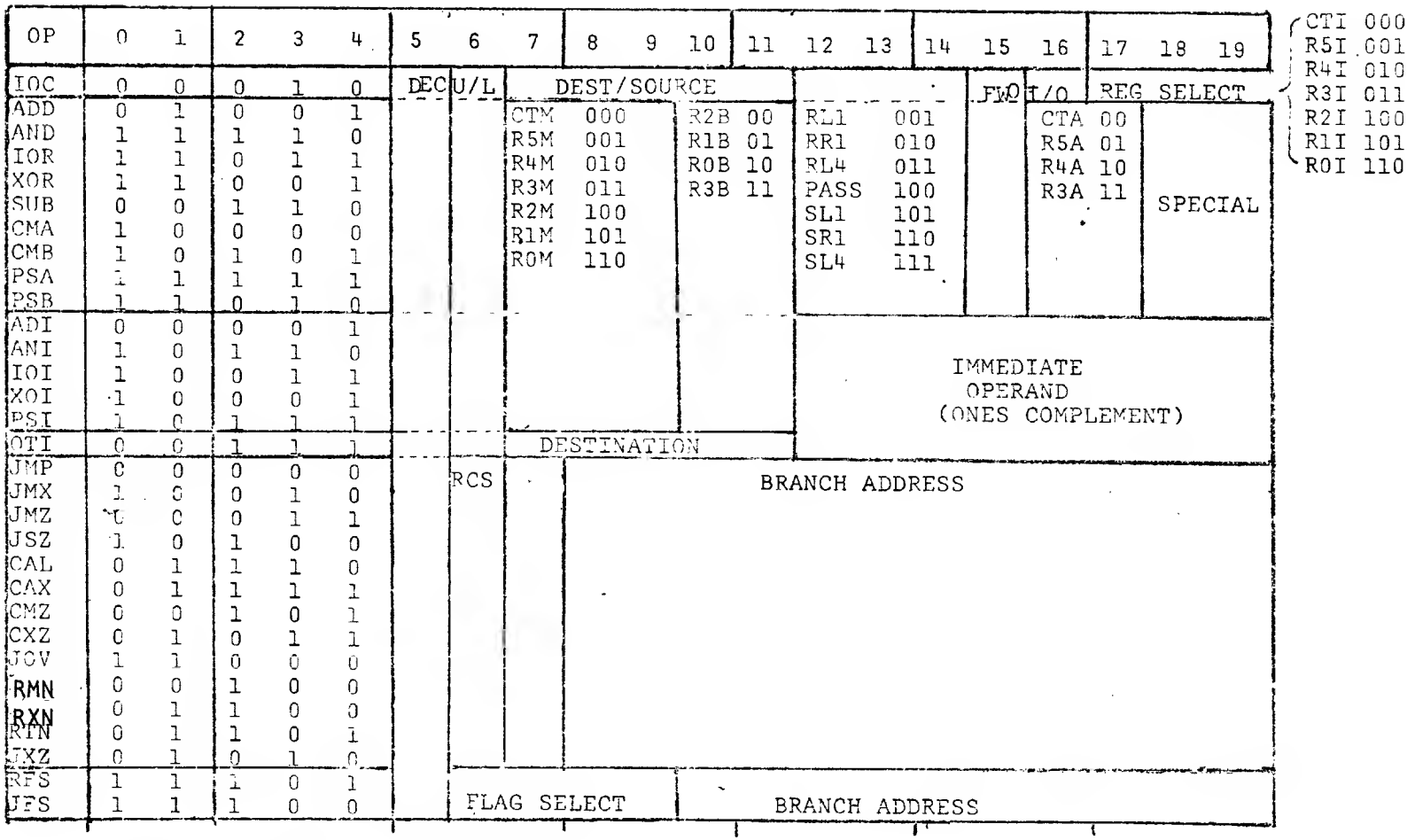
RAR COMPARE OUTPUT (Continued)	It occurs when the processors' ROM Address Register compares with the RAR COMPARE REGISTER. The pulse is positive for one processor timing period (T3, normally 100 nsec.) and has an excursion from approximately +3V to ground (TTL output).
FLAG SWITCH REGISTER	A 16 toggle switch register that will present flags to the processor flag circuitry. The FLAG SW REG jumper must be in the ENABLE position.
FLAG SW REG	A jumper on the Controller Maintenance Panel Interface Board. With the jumper in the INHIBIT position, the FLAG SWITCH REGISTER cannot input to the processor flag circuitry. The jumper in the ENABLE position does allow these inputs and the flag lines from the controller should be disconnected.

APPENDIX C

READ ONLY MEMORY

CONFIGURATION


J2
LSB
MSB
+5V
1K
825 Ω
CS2
CS1
A0
A1
A2
A3
A4
A5
A6
A7
D0
D1
D2
D3
D4
36
33
31
29
27
26
23
21
19
17
15
13
11
9
7
5
3
34
32
30
28



RCS = 0, reverse condition sense
I/O = 0, output
FWO = 0, byte op
DEC = 0, decrement counter
U/L = 0, lower half